CLAIMS

1. Method for the functional verification of a software model (40) of an integrated circuit on demand (ASIC), in a low-level language (for example of the HDL type), which separately handles the generation of the model and the debugging of the functional verification tests to be applied to the model of the circuit for constituting a verification platform, comprising the following two steps:

- creation of an autonomous circuit emulator (1), obtained by replacing the model in a low level (HDL-type) language physically describing the circuit under design to be validated with a high level (for example C++) abstract description generating response data structures in accordance with the functional specification (20) of the project as a function of the stimuli received, this mode being called the "transmission mode."

- integration of the software model (40) in low level (HDL-type) language of the circuit resulting from the design into a verification platform, and creation of the connection of the previously validated autonomous circuit emulator (1), in parallel, to the interfaces of the software model (40) of the circuit, and of the connection of an environment emulator (11, 21, 22); and

- utilization of this platform as a reference for the validation of the response data transmitted by the software model (40) of the circuit, this mode being called the "verification mode.

2. Method according to claim 1, wherein:

- a user generates, using a data processing system, the autonomous simulation configuration (1) corresponding to the software model (40) of the ASIC using the functional specification (20),
- the user writes, from the functional specification (20), and stores in a test platform (21, 22, 23) for integrated circuit models, a program (51) for testing the model (40) of the ASIC, comprising input stimuli sequences to be provided to the software model (40) of the ASIC, which the autonomous simulation configuration (1), based on the functional specification (20), corresponds to output stimuli sequences,

- the user links together, and activates, the autonomous simulation configuration (1) and the test platform (21, 22, 23), and
- he observes the output stimuli of the HDL-type model (40) of the ASIC in order to functionally validate the system constituted by the software model of the ASIC circuit and the validation test program (210), and thus validates the software model (40) in comparison to the functional specification (20) .

3. Method according to either of claims 1 and 2, wherein the autonomous configuration (1) communicates with the user to control the activation of previously created and stored models of input stimuli sequences defined in a high-level programming language, and controls the activation of associated programs (90) for the progressive validation of test sequences determined from the models.

4. Method according to any of claims 1 through 3, wherein the user writes and provides the functional specification (20) in a low-level programming language, specifying functional models of circuits.

5. Method according to any of claims 1 through 4, wherein the user provides the functional specification (20), in the form of a program in a low level language of (HDL type) functional models of circuits, and a program in a high level language of (symbolic (C^{++})) functional models of circuits, and he controls the autonomous simulation configuration (1) so as to perform a co-simulation by synchronizing the execution of the two specification programs.

6. Method according to any of claims 1 through 4, wherein the test platform verifies that the responses of the software model of the ASIC are within response time ranges specified in the functional specification (20).

7. Verification platform for a software model of an integrated circuit on demand (ASIC), characterized in that it comprises data processing means that allow a client to select test models producing input stimuli for the ASIC, these processing means being designed to read functional specification elements (20) of the ASIC and

comprising programs (90) designed to generate a functional validation test program (51) constituted by output stimuli, from the input stimuli and the functional specification elements (20).

8. Verification platform according to claim 7, comprising a library of functional models of circuit blocks for ASICs and means for selecting models through a definition file of the configuration, in order to create a model corresponding to the functional specification of the ASIC that is integrated into the definition of its environment.

9. Verification platform according to either of claims 7 and 8, wherein it is provided, in a link connecting it to the client, with two serial programming language adaptation circuits (11, 12), designed to transform commands in a high level language (C^{++}), used by the client, into commands in a low level (HDL type) language that can be used by the model of the ASIC, and respectively, to transform the commands in a low level language back into commands in a high level language.

10. Verification platform according to any of claims 7 through 9, characterized in that it includes means (90, 10) for executing its operations at the same time as the simulation, which it can interrupt upon detection of an error at the very moment the error appears.

11. Verification platform according to any of claims 7 through 10, characterized in that the functional specification elements (20) are constituted by a truth table or behavior table corresponding to the functions of the various parts or various functional circuit elements of the software model (40) of the ASIC, and the propagation delay ranges to be respected between each input and each output.

12. Verification platform according to any of claims 7 through 11, characterized in that it has a cache memory (962) for storing the blocks used by the nodes according to their addresses, and means for managing, for an address used by one or more nodes, a presence vector with one presence indicator per node.

13. Verification platform according to any of claims 7 through 12, characterized in that the programs (90) are object-oriented and the emulator is structured as a set of classes that makes it possible to manage a collection of execution hypotheses for a transaction in a memory block of the software model, and also to manage transactions that are colliding, i.e., concurrently using the same memory block.

14. Verification platform according to any of claims 7 through 13, characterized in that the algorithms of the programs (90) of the emulator perform the following functions: generating predictions, eliminating predictions, readjusting incorrect predictions, reducing the number of valid hypotheses, and terminating collisions

15. Verification platform according to any of claims 7 through 14, characterized in that it is used as an emulator of a router circuit, a circuit with cache or a router circuit with cache.

16. Verification platform according to any of claims 7 through 15 for testing a software model of an integrated circuit on demand (ASIC), characterized in that it comprises an ASIC emulator (1) for controlling a comparator (23) provided for receiving values generated by the software model of the ASIC circuit tested, upon reception of stimuli sent by at least one stimuli generating circuit (21) storing the test program, an interface (11) for translating the stimuli from an advanced language into a low level language corresponding to that of the software model, and means for validating the verification in case of the detection of a collision by the comparator (23).

17. Verification platform according to any of claims 7 through 16, characterized in that the means for selecting the response to stimuli that depend on the composition of the circuits tested are constituted by a model generated by means for selecting functional models from a library, which associates with each of the

models the responses to a given stimulus, the model corresponding to the composition of the circuit to be tested.

18. Verification platform according to any of claims 7 through 17, characterized in that it includes means (7) for storing the responses thus selected so as to create a test model (70) to be applied to the circuit tested during the reception of stimuli.

19. Verification platform according to any of claims 7 through 18, characterized in that each transaction is constituted, at the level of each interface, by a request packet and one or more associated response packets, wherein the values of the parameters and/or the transmission time constraints of the packets can be forced from the functional test program executed by the emulator of the environment, which appropriately translates all of these parameters during the transmission of the packets to the terminals of the software model of the design.

20. Verification platform according to claim 14, characterized in that the generation of predictions is performed by the emulator of the circuit without having to obtain additional information on the internal operation of the circuit under design.